International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 6, December 2021, pp. 5277~5285
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5277-5285

Using the modified k-mean algorithm with an improved teaching-learning-based optimization algorithm for feedforward neural network training

Morteza Jouyban[1], Mahdie Khorashadizade[2]
[1]Department of Computer Science, Allameh Tabataba'i University, Tehran, Iran
[2]Department of Computer Science, Sistan and Baluchestan University, Zahedan, Iran

Article Info

Article history:

Received Aug 16, 2020
Revised May 19, 2021
Accepted Jun 11, 2021

Keywords:

K-mean clustering
Neural network
Teaching-learning algorithm

ABSTRACT

In this paper we proposed a novel procedure for training a feedforward neural network. The accuracy of artificial neural network outputs after determining the proper structure for each problem depends on choosing the appropriate method for determining the best weights, which is the appropriate training algorithm. If the training algorithm starts from a good starting point, it is several steps closer to achieving global optimization. In this paper, we present an optimization strategy for selecting the initial population and determining the optimal weights with the aim of minimizing neural network error. Teaching-learning-based optimization (TLBO) is a less parametric algorithm rather than other evolutionary algorithms, so it is easier to implement. We have improved this algorithm to increase efficiency and balance between global and local search. The improved teaching-learning-based optimization (ITLBO) algorithm has added the concept of neighborhood to the basic algorithm, which improves the ability of global search. Using an initial population that includes the best cluster centers after clustering with the modified k-mean algorithm also helps the algorithm to achieve global optimum. The results are promising, close to optimal, and better than other approach which we compared our proposed algorithm with them.

Corresponding Author:

Morteza Jouyban
Department of Computer Science
Allameh Tabataba'i University
Tehran, Iran
Email: m.jouyban@gmail.com

1. INTRODUCTION

Artificial neural networks (ANN) are new computational methods and systems for machine learning, knowledge demonstration, and ultimately the application of knowledge to oversee the output of complex systems. The main idea of networks is inspired by the way the biological neural system to process data and information in order to learn and create knowledge. The main philosophy of the artificial neural network is to model the processing characteristics of the human brain to approximate the usual computational methods with the biological processing method. In other words, the artificial neural network is a method that learns the communication knowledge between several sets of data through training and saves it for use in similar cases. This processor works in two ways similar to the human brain: Learning the neural network is done through training, and weighting in the neural network is similar to the information storage system of the human brain. With the help of computer programming knowledge, it is possible to design a data structure that

acts like a neuron. Then the network train by creating a network of these connected artificial neurons, creating a training algorithm for the network, and applying this algorithm to the network. Training the neural network means determining the appropriate weights for the neural network. For this purpose, a strong and fast optimization method should be used. Initially, researchers who were required to train neural networks to solve their problems, generally used from classical neural network training algorithms such as gradient-based optimization like error back-propagation algorithm [1]. But these algorithms, along with their high speed, have major disadvantages, such as the problem of trapping into the local minimum [2]. Another weakness of these algorithms is that they must be applied to a specific and standard category of neural networks such as multilayer perceptron [3]. Although evolutionary algorithms are slightly slower than gradient-based algorithms, they have the ability to escape from local minimum traps, and evolutionary algorithms are not dependent on a specific network structure and can be defined by any network structure. Therefore, many researchers have used evolutionary algorithms to train neural networks [4]-[6]. Therefore, the use of evolutionary algorithms in the neural network has the advantage of escaping from local minimum traps and also not being dependent on a certain network structure.

Clustering is one of the most popular ways to discover data knowledge. The discovery of knowledge is broadly divided into two categories: supervised and unsupervised. A supervised knowledge discovery process typically requires class labels that are sometimes not available in the dataset. This machine model learns using labeled data and having the right answers. Unsupervised machine learning is the process that a machine learns without using labeled data and any teachers. Known discovery knowledge techniques such as clustering can handle unlabeled datasets. The K-means algorithm for clustering is one of the most popular and widely used algorithms [7].

In this paper, after extensive research, we have concluded that the selection of the initial population for artificial neural network training algorithms is very important. So instead of randomly selection an initial population in a small search space, we decided to use result of clustering in a larger search space. Searching in a larger space increases the chances of finding a global optimal or extremely close to optimal answers. Using the modified k-mean algorithm, we divide our large initial population into a limited number of clusters, and then consider the best cluster centers as the initial population of our training algorithm. We then use an improved teaching-learning based optimization algorithm (ITLBO) as the training algorithm. This algorithm solves the problem of trapping in the local optimal and by creating a proper balance between exploration and exploitation, it has been able to get closer to the global optimal.

The rest of this article is being as: In this article, we will first give a brief description of the clustering algorithm used in section 2, and then we present the ITLBO algorithms, and then we describe our proposed method. Then, in section 3, we first provide a brief description of case study datasets, and then we show the results of experiments using the proposed approach applied to the neural network problem. The conclusion of this work is presented in the last section.

2. MATERIALS AND METHODS

2.1. K-mean clustering

Nowadays, clustering is one of the most widely used issues in the field of artificial intelligence. The issue of clustering has particular importance due to the growing volume of web-based texts, textual statues, articles, and can also be effective in improving the results obtained from search engines and information classification. Proper clustering makes it easier to search and access information more efficiently. In general, clustering algorithms can be divided into two general categories: Overlapping clustering method and exclusive clustering method. In the overlapping clustering method, a data can belong to several clusters with different ratios, an example of which is fuzzy clustering. In the exclusive clustering method, after clustering, each data is assigned exactly one cluster, k-mean algorithm belongs to this category. The k-mean clustering is the process of classifying a set of objects into clusters, in which the internal members of each cluster are most similar to each other and have the least similar to members of other clusters. Clustering is the process of separating data or objects into subclasses called clusters. Each cluster contains data that seems to be more similar to each other, and data that appears to be less similar to each other is placed in different clusters.

The k-mean algorithm is one of the most widely used clustering algorithms. This algorithm was first introduced by McQueen in 1967 [8] that is designed for clustering numerical data. In the k-mean algorithm, first the k member is randomly selected from the n members as the cluster centers (k<n). The remaining n-k members are then assigned to the nearest cluster center based on Euclidean distance. After allocating all members, the cluster centers are recalculated by the average value of the clusters members, and this continues until the cluster centers remain stable. Suppose that $D = \{x_1, x_2, \dots, x_n\}$ is a set of n data and $C_1, C_2, \dots, C_k$ are separate clusters on D, in which case the error function is defined is being as:

$$E = \sum_{i=1}^{k} \sum_{x \in c_i} d(x, \mu(c_i)) \quad (1)$$

where μ (c_i) is the center of the c_i cluster and d (x, μ (c_i)) is the distance between x and μ (c_i). Euclidean distances are commonly used to calculate distance in this formula. But in this paper, we use Manhattan distance to identify members belonging to clusters, and we modify the k-mean algorithm in this way. This modified algorithm maintains both numerical and categorical features of the samples. The distance in our method is calculated is being as in (2).

$$d\left(x_j, \mu(c_i)\right) = \frac{\sum_{j=1}^{t} |x_{j,k} - \mu(c_i),_k| + \sum_{j=t+1}^{m} d(x_{j,k} - \mu(c_i),_k)}{|A|} \quad (2)$$

The Manhattan's distance is based on the sum absolute values errors and is less sensitive than the sum squares errors.

2.2. Improving teaching-learning optimization algorithm (ITLBO)

The teaching-learning optimization (TLBO) algorithm was introduced by Rao. In 2011 [9] this algorithm is a popular and powerful optimization algorithm that is used in many engineering and real-world problems. The algorithm is inspired by the process of teaching and learning in a typical classroom.

Although TLBO offers high-quality solutions in the shortest possible time and has consistent convergence [9], in the learning phase of this algorithm, each learner randomly selects another learner from the population. This problem leads to an imbalance between the two concepts of diversity and convergence. ITLBO solves this problem by improving the basic TLBO. In this algorithm, the teaching phase is the same as the basic TLBO algorithm, and the learning phase is described is being as. At this stage, all learners are randomly placed in a rectangular structure. Every learner have to learn from their neighbors. Here, in order to increase the diversity in the algorithm, after a certain number of iterations, the members are randomly rearranged in a rectangular structure. ITLBO has been developed to improve the TLBO algorithm. In TLBO, for example, random choices make low local search abilities, but in ITLBO, along with adding the concept of neighborhood, we try to reduce random choices and use the capabilities of neighbors. This increases the local and global search ability. The main sections of ITLBO are:

2.2.1. ITLBO learning phase

In this phase, each learner is known with an integer and placed in a rectangular array. Neighbors of each learner are clearly identified in Figure 1. At this step, learners may learn from neighbors or the best person in the class. This process is based on local search capability, in addition, a balance is established between local and global search. In a local search, each learner updates their position with the probability of a Pc by the best learner in their neighborhood or by the teacher, the global best, in the population.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Figure 1. A class of learners arranged in a rectangle

$$X_{i,new} = X_{i,old} + r_2.\left(X_{i,teacher} - X_{i,old}\right) + r_3.\left(X_{teacher} - X_{i,old}\right) \quad (3)$$

Where $X_{i,teacher}$ is the teacher in the neighborhood ofX_i and $X_{teacher}$ is the teacher of the whole class, and r_2, r_3 are random numbers in the range of (0, 1). If the new position of each member is improved, the new position will be accepted. In the global search, if the probability of Pc is not met, each learner selects a random learner like X_j from the whole class to provide learning, if X_j is better than X_i, or else, learning establish based on TLBO algorithm learning phase by (4), (5).

$$X_{new} = X_i + r.(X_j - X_i)\ if\ f(X_i) > f(X_j) \quad (4)$$

$$X_{new} = X_i + r.(X_i - X_j)\ if\ f(X_i) < f(X_j) \quad (5)$$

Therefore, using this operation, both local and global search capabilities are obtained. In this algorithm, we improve exploitation capability by this concept that for each person in the population there are a number of neighbors who learn from the best of them. To maintain diversity after a number of iterations, each person's neighboring members change. This makes balance between exploration and exploitation capabilities.

2.3. Investigate the range of variable changes

In evolutionary algorithms, when a new position is obtained for each individual, it may lead to the production of variables that are outside the defined range. In this case, most researchers use from convergence approach based on algorithm 1, but this method is an old and obsolete method that causes the algorithm to trapping in the local minimum. To overcome these problems, we have proposed a new method in algorithm 2 to determine the range of variable changes, as shown in Figure 2. This algorithm prevents the algorithm from trapping in the local optimal and also prevent from making the best answers in high number of iteration.

Algorithm 1. Basic bound constraints handling

```
For j=1→dim
 If X_i(j) > upper bound(j)
 X_i(j) = upper bound(j)
 Else if X_i(j) < lower bound(j)
 X_i(j) = lower bound(j)
 End if
End for
```

(a)

Algorithm 2. Modified bound constraints handling

```
For j=1→dim
 If X_i(j) < lower bound(j)
 X_i(j)=(2* lower bound(j)) −X_i(j)
 Else if X_i(j)>upper bound(j)
 X_i (j)=(2* upper bound(j)) −X_i(j)
 End if
End for
```

(b)

Figure 2. The range of variable changes; (a) algorithm 1: basic bound constraints handling, (b) algorithm 2: modified bound constraints handling

3. THE PROPOSED METHOD

The basic TLBO algorithm, like other evolutionary algorithms, suffers from low convergence speed. In order to improve the performance of the algorithm we modified the learning phase of TLBO algorithm and then we presented ITLBO algorithm with balance between local and global search. And also, we used from a datamining technique called Mk-mean to make more efficient use of the hidden information in the search space to create a more suitable initial population. Clustering means placing data in each cluster that has the least distance and the most similarity.

The importance of initial populations in evolutionary algorithms has long been debated. The initial population is the starting point of any algorithm, and algorithms that start at a proper starting point may have better results than others. In this article, we have tried to start of a stronger initial population by better analyzing the search space. The routine is to use from cluster centers of a large population as initial population instead of randomly select the limited initial population. We consider the best clustering centers as the initial population of our algorithm. The pseudocode of the proposed algorithm for more details is shown in Figure 3.

Algorithm 3. Proposed algorithm

```
Clustering section:
   Input: Dataset D, Number of Clusters k, Dimensions d:
   {C_i is the ith cluster}
   {% Initialization Phase for Clustering algorithm}
   1: (C_1, C_2, ..., C_k)=Initial partition of D.
   {% Iteration Phase for Clustering algorithm}
   2: While (Iter < Max_Iter)
   3: d_ij=distance between case i and cluster j;
   4: n_i= min (d_ij);
   5: Assign case i to cluster n_i;
   6: Re-compute the cluster means of any changed clusters above;
   7: End While
Training section:
   8: Training algorithm population = Clustering Output results.
   {% Initialization Phase for Training algorithm}
   9: Objective function f(X) X = (x_1, x_2, ..., x_d)^t d = number of design variables
   {% Iteration Phase for Training algorithm}
   10: While (Iter < Max_Iter)
   {%Teacher Phase}
   11: Calculate the mean of each design variable X_mean
   12: Identify the best solution (X_teacher)
   13: For i = 1→ pop size
   14: Calculate teaching factor T_F using T_F = round [1 + rand(0,1)]
   15: Modify solution based on best solution (teacher) using X_new = X_i + r.(X_teacher − T_F.X_mean)
   16: Calculate objective function for new mapped student f(X_new)
   17: If X_new is better than X_i
   18: X_i = X_new
   19: End If %End of Teacher Phase
   {% Learner Phase}
   20: finding neighbors for each learner
   21: If rand<P_c
   22: Update the solution using Eq. (3);
   23: Else
   24: Randomly select another learnerX_j, such that j≠i
   25: If X_i is better than X_j
   26: Update the solution using Eq. (5);
   27: Else
   28: Update the solution using Eq. (4);
   29: End If
   30: If X_new is better than X_i
   31: X_i = X_new
   32: End If
   33: End If
   29: End For
   30: End While
```

Figure 3. Suggested method

4. EXPERIMENTS

4.1. Defining classification problems and predicting time series

In this section, we evaluate the effectiveness of the proposed method using ten classification problems and two time series prediction problems, and to prove the effectiveness of the proposed algorithm, its results are compared with other algorithms such as basic TLBO and ITLBO. In addition, the results of the improved training algorithm have been compared with the basic neural network training algorithms from other articles, and then in another step, the performance of the proposed method has been compared with other methods available in the research literature, the following experiments can be seen in detail. Classification problems include Iris, diabetes diagnosis, thyroid disease, breast cancer, credit card, glass, heart, wine, page blocks, and liver disorders. Time series prediction problems include Mackey-Glass [10] and gas furnaces [11]. The number of features in the classification problem, the number of classes and the total number of samples listed in Table 1. Classification problems are taken from the UCI machine learning repository [12]. The Mackey-Glass is a dataset that obtained from the (6), which is td=17.

$$\frac{dx(t)}{dt} = -bx(t) + \frac{ax(t-t_d)}{1+x^{10}(t-t_d)} \quad (6)$$

For the Mackey-Glass dataset, we considered the x (t+6) output with the input variables x(t), x(t-6), x(t-12) and x(t-18). For gas furnace dataset, the input variables are u(t-3), u(t-2), u(t -1), y(t-3), y(t-2), y(t-1) and the output variable is y(t), as reported in previous works [13]. We implemented this algorithm by MATLAB. We used 30 runs to evaluate the performance of this model. The datasets were randomly divided into two sets: The training set and the test set for each run. 70% of the total data was used for the training set and the rest of the data was used as a test set to examine the model. Datasets are normalized to the interval [-1], [1] using the min-max normalization method.

Table 1. Explain the datasets used for the proposed method

Dataset	Instances	Features	Classes
1. Iris	150	4	3
2. Diabetes	768	8	2
3. Thyroid	7200	21	3
4. Cancer	699	10	2
5. Card	690	15	2
6. Glass	214	10	6
7. Heart	270	13	2
8. Wine	178	13	3
9. Page-blocks	5473	10	5
10. Liver	345	6	2
11. Mackey-Glass	1000	1	0
12. Gas Furnace	296	2	0

4.2. The results of the comparison of the proposed methods

The performance of the algorithms evaluated and compared based on two criteria, training, and testing errors that on the classification problems it means classification error percentage. The error function for Mackey-Glass dataset is root mean squared error (RMSE) and for gas furnace dataset is MSE. The average results after 30 runs for the algorithms are shown in Table 2. The results show the superiority of the proposed method. It can be seen that the MK-ITLBO algorithm performs better than other methods. We used the average ranking test to find the best algorithm. These results are found using the RANK function in Microsoft Excel, and the average rankings are shown in Table 3. The results show that MK-ITLBO ranks first in all cases for training errors and testing errors.

Table 2. Average ranking for the proposed algorithms

Training error	TLBO	ITLBO	MK-ITLBO	Algorithm
	2.7500	2.1667	1.0833	Rank
Testing error	TLBO	ITLBO	MK-ITLBO	Algorithm
	2.9167	2.0000	1.0833	Rank

Table 3. P-value results for pairwise comparing of MK-ITLBO versus other algorithms by Wilcoxon test

TLBO	ITLBO	Criteria	Dataset
1. Iris	Training error	2.0056e-03	5.2453e-05
	Testing error	3.7446e-04	2.9940e-05
2. Diabetes	Training error	2.9707e-05	2.5444e-08
	Testing error	1.5884e-04	1.0405e-09
3. Thyroid	Training error	2.9027e-11	9.5982e-10
	Testing error	3.6205e-11	2.9082e-11
4. Cancer	Training error	5.5569e-04	3.0671e-09
	Testing error	3.0050e-07	2.6706e-11
5. Card	Training error	1.3054e-05	7.8263e-08
	Testing error	1.8508e-10	6.5277e-10
6. Glass	Training error	2.1957e-02	6.3243e-08
	Testing error	9.9797e-10	2.5432e-11
7. Heart	Training error	3.7816e-05	4.6398e-06
8. Wine	Testing error	7.5102e-05	2.6689e-11
	Training error	3.3801e-01	3.3727e-02
9. Page-blocks	Testing error	2.4009e-08	2.5676e-11
	Training error	3.4339e-01	1.6744e-01
10. Liver	Testing error	9.9407e-01	2.9860e-11
	Training error	5.1138e-05	2.4198e-01
11. MackeyGlass	Testing error	5.2746e-02	1.3604e-11
	Training error	1.6743e-04	7.0507e-09
12. Gas Furnace	Testing error	3.3079e-02	7.0507e-09
	Training error	5.8282e-03	4.3349e-07
	Testing error	1.5030e-02	3.0199e-11

To evaluation whether the MK-ITLBO result was significantly better than other algorithms, we calculated the p-value test for all datasets with a significant level of 0.05 for datasets. From a statistical perspective p-value, the probability of rejecting the null hypothesis with the accuracy conditions is based on the observed data. The smaller the probability, the greater our confidence in the reality of the observed difference. The P-values calculated for MK-ITLBO compared to other algorithms are shown in Table 4. Table 5 compares our proposed MK-ITLBO algorithm and other approaches for classification problems.

Table 4. Comparison of MK-ITLBO and other approaches for classification problems

Dataset	MK-ITLBO	T-LogisticBatDN	SA	TS	GA	PSO	GaTSa+BP
Iris	0	5.2632	12.649	12.478	2.5641	4.6154	5.2564
Diabetes	18.0698	3.1579	27.156	27.404	25.994	25.876	27.061
Thyroid	5.0825	22.0164	7.3813	7.3406	7.2850	7.3322	7.1509
Cancer	2.1643	5.2222	7.1729	7.2779	7.4220	6.2846	15.242
Card	11.8428	6.9425	23.469	18.042	31.724	21.269	15.242
Glass	20.0535	6.4423	58.381	56.412	58.031	57.777	55.142

Table 5. Comparing the results of best algorithm with other methods in literature

		Article number (approach)												
Dataset	Criteria	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	Mk-ITLBO
Iris	Tr_E (%) Te_E (%)	- -	- 1.33	1.79 1.95	- 3.73	1.53 1.69	-	2.3 2.2	- 2.67	2.7 4.87	0.88 4.37	3.33 -	- 4.61	0 0
Diabetes	Tr_E (%) Te_E (%)	- 21.66	-	14.96 16.63	- 23.09	13.96 15.63	-	21.6 21.8	- 23.95	-	[26] - 29.87		- 25.87	16.197 18.0698
Thyroid	Tr_E (%) Te_E (%)	- -	3.18	5.27 5.41	-	5.47 5.11	-	-	-	-	[26] - 9.72	- 6.05	- 7.32	5.0519 5.0825
Cancer	Tr_E (%) Te_E (%)	- 2.59	-	2.95 3.02	- 3.13	2.35 2.45	- 5.34	2.6 2.3	- 2.36	2.27 2.39	-	-	-	1.6182 2.1643
Card	Tr_E (%) Te_E (%)	-	-	12.90 13.14	-	12.57 12.58	-	-	-	-	[26] - 5.56	- 7.97	-	11.5556 11.8428
Wine	Tr_E (%) Te_E (%)	- 0.48	- 5.27	-	- 2.81	-	- 3.95	1.6 0.3	- 1.11	1.55 6.3	[24] 3.92 4.66	- 2.81	-	0.54201 3.0424
Heart	Tr_E (%) Te_E (%)	- 13.56	- 15.55	-	- 40.88	-	-	- 16.6	-	-	-	- 15.74	-	10.4233 12.5255
Liver	Tr_E (%) Te_E (%)	- 27.98	-	-	-	-	-	-	24.28	-	-	20.87	-	20.7451 22.9231
page-blocks	Tr_E (%) Te_E (%)	-	-	-	-	-	- 3.92	-	-	-	[26] - 5.56	-	-	5.8544 6.3727
Glass	Tr_E (%) Te_E (%)	- 31.23	-	33.05 34.09	38.94 -	26.05 28.09	- 26.5	- -	- 25.6	-	-	- 26.17	- -	16.2387 20.0535
Mackey-Glass	Tr_E (%) Te_E (%)	-	[27] 5.5E-04 4.7E-04	1.6E-05 1.7E-05	-	0.001 0.001	-	-	-	-	-	-	- 0.68	2.1514e-4 2.4224e-4
Gas Furnace	Tr_E (%) Te_E (%)	-	[27] 0.003 0.004	0.26 0.28	-	0.18 0.19	-	-	-	-	-	-	-	6.1936e-4 7.9393e-4

This table shows the results of our proposed method with some datasets that have been selected as a sample. The results represent that the proposed method is superior to other basic neural network training algorithms.

4.3. The results compare the proposed combined method with other methods available in the literature

In this section, we compare the top algorithm among the proposed hybrid algorithms with other methods available in the literature and make all comparisons with the dataset sets introduced in this paper. Table 5 compares the MK_ITLBO method with the approaches introduced above. Fields with a symbol- indicate that the proposed approach did not work on that dataset or that the results are not available. In this table Tr_E (%) means training error percentage and Te_E (%) means testing error percentage.

5. CONCLUSION

This article presented an improved teaching-learning based optimization algorithm for neural network training. We used two methods to improve the performance of the basic algorithm. First, to make a more effective search space for the initial population of the algorithm, instead of randomly selecting the population we used result of clustering with the modified k-mean algorithm, and second, we have improved the teaching-learning optimization algorithm to create a balance between exploitation and exploration. We have applied our method to classification and time series prediction problems. The results in section 4 show the superior performance of the proposed algorithm, as mentioned, this version has a powerful training algorithm against premature convergence that also balances exploitation and exploration. In addition, it is combined with the Mk-mean algorithm, which examines the search space more effectively. We also confirmed these results with statistical tests, and then compared this algorithm with other methods of literature, and based on different evaluations it was concluded that this algorithm has a better ability than other algorithms regarding classification and time series prediction errors. The results motivate us to find approach to change our method to the future works. This development could be in the use of chaotic mapping in the method.

REFERENCES

[1] N. Zhang, "An online gradient method with momentum for two-layer feedforward neural networks," *Applied Mathematics Computation*, vol. 212, no. 2, pp. 488-498, 2009, doi: 10.1016/j.amc.2009.02.038.
[2] M. Gori and A. Tesi, "On the problem of local minima in backpropagation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 76-86, Jan. 1992, doi: 10.1109/34.107014.
[3] O. Aran and E. Alpaydın, "An Incremental Neural Network Construction Algorithm for Training Multilayer Perceptrons," *Artificial Neural Networks and Neural Information Processing*, Istanbul, Turkey: ICANN/ICONIP 2003.
[4] S. Kulluk, L. Ozbakir, and A. Baykasoglu, "Training neural networks with harmony search algorithms for classification problems," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 1. pp. 11-19, Feb. 2012, doi: 10.1016/j.engappai.2011.07.006.
[5] N. S. Jaddi, S. Abdullah, and A. R. Hamdan, "Optimization of neural network model using modified bat-inspired," *Applied Soft Computing*, vol. 37, pp. 71-86, Dec. 2015, doi: 10.1016/j.asoc.2015.08.002.
[6] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)*, 2003, pp. 110-117, doi: 10.1109/SIS.2003.1202255.
[7] K. Krishna and M. N. Murty, "Genetic K-means algorithm," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433-439, Jun. 1999, doi: 10.1109/3477.764879.
[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, pp. 281-297, Jun. 1967.
[9] R. V. Rao and V. J. S. D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303-315, Mar. 2011, doi: 10.1016/j.cad.2010.12.015.
[10] C. H. López-Caraballo, Salfate, J. A. Lazzús, P. Rojas, M. Rivera, and L. Palma-Chilla, "Mackey-Glass noisy chaotic time series prediction by a swarm-optimized neural network," *Journal of Physics: Conference Series. IOP Publishing*, vol. 720, no. 1, pp. 12002-12011, May 2016, doi: 10.1088/1742-6596/720/1/012002.
[11] K. Dunn, "The gas furnace data set from Box and Jenkins Book on Time Series Analysis," OpenMV.net Datasets, 2018. [Online]. Available: http://openmv.net/info/gas-furnace
[12] C. Blake, "UCI Repository of Machine Learning Databases," University of California at Irvine, 1998.
[13] K. Yu, X. Wang, and Z. Wang, "Constrained optimization based on improved teaching-learning-based optimization algorithm," *Information Sciences*, vol. 352-353, pp. 61-78, Jul. 2016, doi: 10.1016/j.ins.2016.02.054.
[14] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "META-DES. Oracle: Meta-learning and feature selection for dynamic ensemble selection," *Information Fusion*, vol. 38, pp. 84-103, 2017, doi: 10.1016/j.inffus.2017.02.010.

[15] H. Wang and Z. Zhou, "An improved rough margin-based ν-twin bounded support vector machine," *Knowledge-Based Systems*, vol. 128, pp. 125-138, Jul. 2017, doi: 10.1016/j.knosys.2017.05.004.
[16] N. S. Jaddi and S. Abdullah, "A cooperative-competitive master-slave global-best harmony search for ANN optimization and water-quality prediction," *Applied Soft Computing*, vol. 51, pp. 209-224, Feb. 2017, doi: 10.1016/j.asoc.2016.12.011.
[17] D. P. F. Cruz, R. D. Maia, L. A. da Silva, and L. N. Castro, "BeeRBF: a bee-inspired data clustering approach to design RBF neural network classifiers," *Neurocomputing*, vol. 172, pp. 427-437, Jan. 2016, doi: 10.1016/j.neucom.2015.03.106.
[18] N. S. Jaddi, S. Abdullah, and M. A. Malek, "Master-Leader-Slave Cuckoo Search with Parameter Control for ANN Optimization and Its Real-World Application to Water Quality Prediction," *PLoS ONE*, vol. 12, no. 1, 2017, Art. no. e0170372, doi: 10.1371/journal.pone.0170372.
[19] M. F. Mohammeda and C. P. Lim, "Improving the Fuzzy Min-Max neural network with a K-nearest hyperbox expansion rule for pattern classification," *Applied Soft Computing*, vol. 52, pp. 135-145, Mar. 2017, doi: 10.1016/j.asoc.2016.12.001.
[20] B. Y. Hiew, S. C. Tan, and W. S. Limb, "A double-elimination-tournament-based competitive co-evolutionary artificial neural network classifier," *Neurocomputing*, vol. 249, pp. 345-356, Aug. 2017, doi: 10.1016/j.neucom.2016.11.082.
[21] K. Sun, J. Zhang, C. Zhang, and J. Hu, "Generalized extreme learning machine autoencoder and a new deep neural network," *Neurocomputing*, vol. 230, pp. 374-381, Mar. 2017, doi: 10.1016/j.neucom.2016.12.027.
[22] Z. Gao, C. Ma, D. Song, and Y. Liu, "Deep quantum inspired neural network with application to aircraft fuel system fault diagnosis," *Neurocomputing*, vol. 238, pp. 13-23, May 2017, doi: 10.1016/j.neucom.2017.01.032.
[23] J. Wang, Q. Cai, Q. Chang, and J. M. Zurada, "Convergence analyses on sparse feedforward neural networks via group lasso regularization," *Information Sciences*, vol. 381, pp. 250-269, Mar. 2017, doi: 10.1016/j.ins.2016.11.020.
[24] S. Shinde and U. Kulkarni, "Extended fuzzy hyperline-segment neural network with classification rule extraction," *Neurocomputing*, vol. 260, pp. 79-91, Oct. 2017, doi: 10.1016/j.neucom.2017.03.036.
[25] T. B. Ludermir, A. Yamazaki, and C. Zanchettin, "An Optimization Methodology for Neural Network Weights and Architectures," in *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1452-1459, Nov. 2006, doi: 10.1109/TNN.2006.881047.
[26] C. Zhanga, C. Liu, X. Zhang, and G. Almpanidis, "An up-to-date comparison of state-of-the-art classification algorithms," *Expert Systems with Applications*, vol. 82, pp. 128-150, Oct. 2017, doi: 10.1016/j.eswa.2017.04.003.
[27] N. S. Jaddi, S. Abdullah, and A. R. Hamdan, "A solution representation of genetic algorithm for neural network weights and structure," *Information Processing Letters*, vol. 116, no. 1, pp. 22-25, Jan. 2016, doi: 10.1016/j.ipl.2015.08.001.